

Durham Research Online

Deposited in DRO:

31 August 2016

Version of attached file:

Accepted Version

Peer-review status of attached file:

Not peer-reviewed

Citation for published item:

Bennett, K.H. and Rajlich, V.T. (2001) 'Software evolution : a road map.', in Proceedings, IEEE International Conference on Software Maintenance : systems and software evolution in the era of the Internet : Florence, Italy, 7-9 November, 2001. Los Alamitos, CA: IEEE Computer Society Press, p. 4.

Further information on publisher's website:

<http://dx.doi.org/10.1109/ICSM.2001.972703>

Publisher's copyright statement:

© 2001 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Additional information:

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full DRO policy](#) for further details.

Software Evolution: A Road Map

K. H. Bennett
Research Institute for Software Evolution
University of Durham
UK
DH1 3LE
keith.bennett@durham.ac.uk

V. T. Rajlich
Department of Computer Science
Wayne State University
Detroit, MI 48202
vtr@cs.wayne.edu

Abstract

The aim of the panel session is to clarify issues involved in software evolution, but more particularly, to identify and explore promising solutions and research directions, in order to identify strategic goals for the research community and solution visions for practitioners. By evolution, we mean a stage of software lifecycle where the software undergoes substantial changes and this also includes iterative software development and/or agile software processes.

Evolution methodology

In broad terms, two types of solution are emerging as promising ways forward. The first may be termed *anticipatory*, and is based on the idea that processes and product evolution can both be planned, and preparation made for changes before they actually occur. This is the traditional view of the waterfall lifecycle and software engineering textbooks. The second is termed *reactive*, and offers a much more opportunistic view: that software change is too unpredictable to be planned, and we must put in place the right mechanisms to make changes when and as they are demanded. In both cases, there is increasing empirical evidence available from real-world software maintenance projects to inform the debate. Also in both cases, we have radically different perspectives of the software lifecycle that is employed.

Speed of evolution

Ultimately, we must meet the demands and needs of our users, who are increasingly demanding faster time to market as the key competitive edge. So we must accomplish evolutionary software change, not in the months or weeks that have been the norm, but in days

or, better, hours. Businesses involved in the Internet are having to go constant urgent change, continually reinventing themselves to meet new challenges. This *emergent* organisational behaviour poses many new practical and research problems for the maintenance community.

Panel format

The panel will address this by inviting a team of distinguished and experienced software engineers to consider two key questions:

Is the best way forward for software evolution to concentrate on the reactive or anticipatory approach?

Can we attain a better understanding of evolution and meet the demands of our users, many of whom operate in emergent organisations and need very rapid software changes?